

Uma *factory function* (“função fábrica”) é como chamamos, em JavaScript, uma função que retorna um objeto.

Exemplo de *factory function*:

```
function criaUser(nome, email) {  
  return {  
    nome,  
    email,  
    exibeInfos() {  
      return `${nome}, ${email}`  
    }  
  }  
}
```

COPIAR CÓDIGO

Veja que, um pouco diferente do que foi feito na aula, não se trata de um objeto literal mas sim uma função que **retorna** (através da palavra-chave `return`) um objeto.

A função acima, quando executada com os parâmetros necessário, vai retornar um objeto com duas propriedades (`nome` e `email`) e um método interno (`exibeInfos()`):

```
const newUser = criaUser('Rodrigo', 'r@r.com')  
console.log(newUser)
```

```
console.log(newUser.exibeInfos())
```

COPIAR CÓDIGO

O retorno esperado no terminal:

```
{  
  nome: 'Rodrigo',  
  email: 'r@r.com',  
  exibeInfos: [Function: exibeInfos]  
}
```

Rodrigo, r@r.com

COPIAR CÓDIGO

As *factory functions* são diferentes das funções construtoras. Veja um exemplo de função construtora, como fizemos anteriormente durante a aula:

```
function User(nome, email) {  
  this.nome = nome  
  this.email = email  
  
  this.exibeInfos() {  
    return `${nome}, ${email}`  
  }  
}
```

COPIAR CÓDIGO

As funções construtoras devem ser chamadas com o operador `new` para criar *instâncias* do objeto `User`:

```
const newUser = new User('Mariana', 'm@m.com')  
console.log(newUser)
```

```
console.log(newUser.exibeInfos())
```

COPIAR CÓDIGO

O retorno no console é praticamente o mesmo, com exceção que agora `exibeInfos()` é uma função anônima e o objeto é explicitamente uma instância de `User`:

```
User {  
  nome: 'Mariana',  
  email: 'm@m.com',  
  exibeInfos: [Function (anonymous)]  
}
```

Mariana, m@m.com [COPIAR CÓDIGO](#)

No caso da *factory function* não há perda de contexto na execução dos métodos internos.

Quais as diferenças e qual devo usar?

Não há consenso, pois ambas têm vantagens e desvantagens. A sintaxe da função construtora é mais confortável para quem está acostumado com o uso de classes; já a *factory function* é mais flexível a respeito do tipo de objeto que será retornado.

Boa parte dos frameworks e bibliotecas do NodeJS vão trabalhar com classes ou construtores, instanciados com `new`.